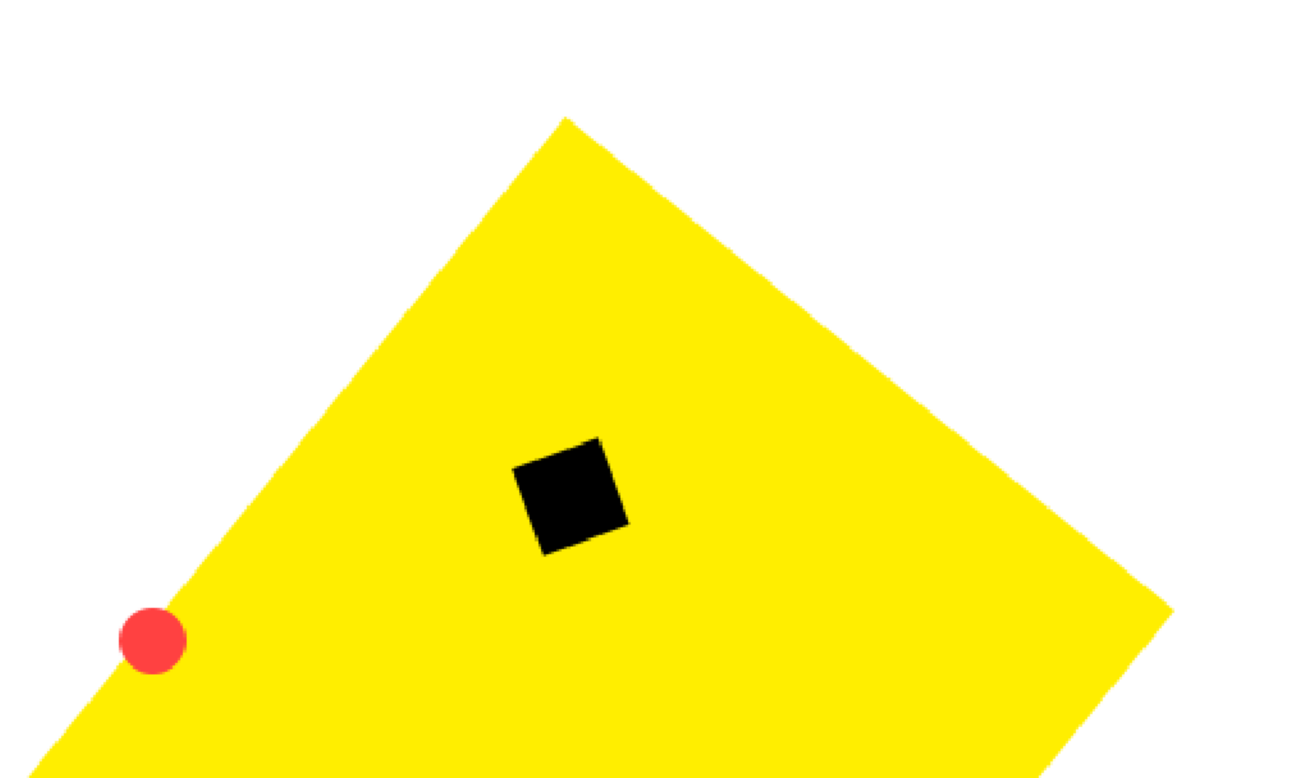
Готовим тестовые данные

Захаров Дмитрий





Обо мне

Дмитрий Захаров

- Работаю фронтенд-разработчиком в Росбанке

Обо мне

Дмитрий Захаров

- Работаю фронтенд-разработчиком в Росбанке
- В IT с 2010 года

Обо мне

Дмитрий Захаров

- Работаю фронтенд-разработчиком в Росбанке
- В IT с 2010 года
- Успел поработать в разных ролях: сисадмин / DevOps, тестировщик, разработчик

Обо мне

Дмитрий Захаров

- Работаю фронтенд-разработчиком в Росбанке
- В IT с 2010 года
- Успел поработать в разных ролях: сисадмин / DevOps, тестировщик, разработчик
- Периодически контрибьючу в опенсорс: nestjs, grpc-web, @nrwl/nx, angular-eslint, falso и т.д.

Повестка дня

(Не та, о которой вы подумали)

- Разберем кейсы применения тестовых данных

Повестка дня

(Не та, о которой вы подумали)

- Разберем кейсы применения тестовых данных
- Поговорим о проблемах взаимодействия между фронтами и бэкендерами

Повестка дня

(Не та, о которой вы подумали)

- Разберем кейсы применения тестовых данных
- Поговорим о проблемах взаимодействия между фронтами и бэкендерами
- Разберем инструменты для генерирования тестовых данных

Повестка дня

(Не та, о которой вы подумали)

- Разберем кейсы применения тестовых данных
- Поговорим о проблемах взаимодействия между фронтами и бэкендерами
- Разберем инструменты для генерирования тестовых данных
- Рассмотрим плюсы и минусы разных подходов

Подготовка тестовых данных

Подготовка тестовых данных

Разберем кейсы, где актуально подготовить тестовые данные

- Когда разрабатываем параллельно с бэкендом, и нет данных

Подготовка тестовых данных

Разберем кейсы, где актуально подготовить тестовые данные

- Когда разрабатываем параллельно с бэкендом, и нет данных
- Когда показываем результаты работы еще до интеграции с бэком

Подготовка тестовых данных

Разберем кейсы, где актуально подготовить тестовые данные

- Когда разрабатываем параллельно с бэкендом, и нет данных
- Когда показываем результаты работы еще до интеграции с бэком
- Когда пишем e2e или интеграционные тесты

Подготовка тестовых данных

Разберем кейсы, где актуально подготовить тестовые данные

- Когда разрабатываем параллельно с бэкендом, и нет данных
- Когда показываем результаты работы еще до интеграции с бэком
- Когда пишем e2e или интеграционные тесты
- Когда у вас есть деморежим в вашем приложении

Типы тестовых данных

Типы тестовых данных

- Явные (постоянные aka хард-код)

Типы тестовых данных

- Явные (постоянные aka хард-код)
- Рандомные

Когда какой тип применять?

Явные тестовые данные

- Unit-тесты

Явные тестовые данные

- Unit-тесты
- Edge case’ы, где важна точность

Явные тестовые данные

pipe

```
import { BooleanTextPipe } from './boolean-text.pipe';

describe('BooleanPipe', () => {

  it('should be Yes when true provided', () => {
    const pipe = new BooleanTextPipe();

    expect(pipe.transform(true)).toEqual('Да');
  });

  it('should be Yes when 1 provided', () => {
    const pipe = new BooleanTextPipe();

    expect(pipe.transform(1)).toEqual('Да');
  });

  it('should be No when false provided', () => {
    const pipe = new BooleanTextPipe();

    expect(pipe.transform(false)).toEqual('Нет');
  });
});
```

Рандомные тестовые данные

- Интеграционные / e2e-тесты

Рандомные тестовые данные

- Интеграционные / e2e-тесты
- Для показа результатов выполненной фичи

Рандомные тестовые данные

- Интеграционные / e2e-тесты
- Для показа результатов выполненной фичи
- Для деморежима приложения

Рандомные данные

Подготовка тестовых данных

А какие же тестовые данные нам нужны?

- Понятные для человека
- Близкие к реальности
- Корректные по бизнес-смыслу
- Гибкие в плане отображения локали (рус / англ / и т.д.)

Тестовые данные

Рублёвые платежи

Импорт | Создать платеж

Исходящие | Входящие | Шаблоны | Картотека

	№	Дата	Контрагент	Назначение	Сумма	Статус	
☐	85063	04.10.2022	Dooley and Sons 40302.450.8.79538886425	Оплата по договору 2354-3	8 224 203.97 RUB	Исполнен	•••
☐	1571	03.10.2022	Weber Inc 40302.714.3.15645208888	Оплата по договору 2354-1	621 345.91 RUB	Создан	•••
☐	54018	02.10.2022	Weber Inc 40302.563.1.85823432403	Оплата за материалы по договору №25-Р от 29.03.2020 и накладной №156 от...	3 103 236.22 RUB	Требуется подтверждение по СМС	•••
☐	90633	30.09.2022	Welch, Lockman and Hand 40302.717.1.47470448786	Оплата по договору 2353 НДС не облагается	2 202 192.23 RUB	Черновик	•••
☐	63624	30.09.2022	Kuhlman, Schowalter and West 26468.465.6.87605469343	Предварительная оплата за транспортные услуги по счёту № 12 от...	6 477 357.93 RUB	Создан	•••
☐	48703	29.09.2022	Jacobi - Kutch 40302.874.3.94423783487	Оплата по договору 2354-3	3 150 731.67 RUB	Отменен	•••
☐	84122	28.09.2022	Steuber, Luettgen and Corkery 72861.638.0.84316833175	Оказание услуг по договору 432 на основании акта № 25 от 25 июня 2020...	2 904 416.21 RUB	В работе	•••

Тестовые данные

person

```
{
  "firstName": "Дмитрий",
  "lastName": "Захаров",
  "addresses": [
    {
      "city": "Москва",
      "street": "ул. Рандомная, д.1",
      "appartment": "33"
    }
  ]
}
```

Подготовка тестовых данных

Какие есть проблемы?

- Явный хард-код

Подготовка тестовых данных

Какие есть проблемы?

- Явный хард-код
- Только одна локаль

Решение: использовать библиотеки для генерации тестовых данных

Применение на практике

Подготовка тестовых данных

- Когда разрабатываем параллельно с бэкендом, и нет данных
- Когда показываем результаты работы еще до интеграции с бэком
- Когда пишем e2e или интеграционные тесты
- Когда у вас есть деморежим в вашем приложении

Взаимодействие

Взаимодействие

Какие бывают проблемы?

- Мы берем задачи только после того, как готов бэк

Взаимодействие

Какие бывают проблемы?

- Мы берем задачи только после того, как готов бэк
- Проблема структур данных (бэкендер и фронтендер смотрят со своих колоколен)

Взаимодействие

Какие бывают проблемы?

response

```
{
 "packages": [
   {
     "firstName": "Дмитрий",
     "lastName": "Захаров",
     "address": "Москва, ул. Рандомная 1, д.1, 33"
   }
 ]
}
```

Взаимодействие

Какие бывают проблемы?

response

```
[
  {
     "firstName": "Дмитрий",
     "lastName": "Захаров",
     "addresses": [
       {
         "city": "Москва",
         "street": "ул. Рандомная 1, д.1",
         "appartment": "33"
       }
     ]
  }
]
```

Взаимодействие

Какие бывают проблемы?

- Мы берем задачи только после того, как готов бэк
- Проблема структур данных (бэкендер и фронтендер смотрят со своих колоколен)
- Опечатки в названиях или “кривые” переводы на англ

Взаимодействие

Какие бывают проблемы?

response

```
[
  {
    "firstName": "Дмитрий",
    "lostName": "Захаров",
    "addresses": [
      {
        "city": "Москва",
        "street": "ул. Рандомная 1, д.1",
        "appartment": "33"
      }
    ],
    "price": 1042,
    "currency": "Russian Rubbles"
  }
]
```

Решение

Нужно больше общения и contract first-подход

- Собраться на двоих-троих и обсудить договоренности (в том числе утвердить все в задаче как Definition of Done)

Решение

Нужно больше общения и contract first-подход

- Собраться на двоих-троих и обсудить договоренности (в том числе утвердить все в задаче как Definition of Done)
- Накидать драфт структуры данных и договориться так, чтоб всех устраивало

Решение

Нужно больше общения и contract first-подход

- Собраться на двоих-троих и обсудить договоренности (в том числе утвердить все в задаче как Definition of Done)
- Накидать драфт структуры данных и договориться так, чтоб всех устраивало
- Проверить опечатки, корректность переводов с русского на англ в именовании и т.д.

Решение

Нужно больше общения и contract first-подход

- Собраться на двоих-троих и обсудить договоренности (в том числе утвердить все в задаче как Definition of Done)
- Накидать драфт структуры данных и договориться так, чтоб всех устраивало
- Проверить опечатки, корректность переводов с русского на англ в именовании и т.д.
- Три Амиго

Три Амиго



Три Амиго

- Бизнес: какую проблему мы пытаемся решить?
- Разработка: как мы можем создать решение для этой проблемы?
- Тестирование: что может пойти не так?

Доклад Глеба Михеева про Contract First подход



Детали процесса разработки

Детали процесса разработки

- Создаем контракт вместе с бэкендером (json / interface / proto)

Детали процесса разработки

- Создаем контракт вместе с бэкендером (json / interface / proto)
- Создаем у себя TypeScript класс или интерфейс для модели данных

Детали процесса

Структура данных

Statement

```
/**
 * Шапка выписки
 */
export interface StatementHeader {
    /**
     *  Название группы счетов
     */
    accountGroupName?: string;
    /**
     *  Дата начала выписки
     */
    dateFrom?: string;
    /**
     *  Дата конца выписки
     */
    dateTo?: string;
    incomingBalance?: Money;
    outgoingBalance?: Money;
    /**
     *  Статус (создан, обработка и т.д.)
     */
    status?: StatementHeaderStatusEnum;
    turnoversCredit?: Money;
    turnoversDebit?: Money;
}
```

Детали процесса разработки

- Создаем контракт вместе с бэкендером (json / interface / proto)
- Создаем у себя TypeScript класс или интерфейс для модели данных
- Внедряем тестовые данные в стор (эффекты), или сервис

Детали процесса разработки

- Создаем контракт вместе с бэкендером (json / interface / proto)
- Создаем у себя TypeScript класс или интерфейс для модели данных
- Внедряем тестовые данные в стор (эффекты), или сервис
- Выполняем разработку так, словно данные приходят с бэкенда

Что получаем в итоге?

Переделать потом — дороже, чем сейчас

(c) опыт

Результаты

Переделать потом — дороже, чем сейчас

- Низкий шанс переделок

Результаты

Переделать потом — дороже, чем сейчас

- Низкий шанс переделок
- Рассмотрение контракта с обоих углов (фронт / бэк)

Результаты

Переделать потом — дороже, чем сейчас

- Низкий шанс переделок
- Рассмотрение контракта с обоих углов (фронт / бэк)
- Закрепление контракта (как минимум готова структура данных) либо swagger / grpc proto

Результаты

Переделать потом — дороже, чем сейчас

- Низкий шанс переделок
- Рассмотрение контракта с обоих углов (фронт / бэк)
- Закрепление контракта (как минимум готова структура данных) либо swagger / grpc proto
- Возможность параллельной разработки вместе с бэкендером

Результаты

Переделать потом — дороже, чем сейчас

- Низкий шанс переделок
- Рассмотрение контракта с обоих углов (фронт / бэк)
- Закрепление контракта (как минимум готова структура данных) либо swagger / grpc proto
- Возможность параллельной разработки вместе с бэкендером
- Возможность показать результат с помощью моковых тестовых данных тем, кто принимает результат работы

Результаты

Переделать потом — дороже, чем сейчас

- Низкий шанс переделок
- Рассмотрение контракта с обоих углов (фронт / бэк)
- Закрепление контракта (как минимум готова структура данных) либо swagger / grpc proto
- Возможность параллельной разработки вместе с бэкендером
- Возможность показать результат с помощью моковых тестовых данных тем, кто принимает результат работы
- Меньше изменений в коде при реальной интеграции с бэком

Результаты

Переделать потом — дороже, чем сейчас

- Бонус: можно покрывать юнит-тестами

Инструменты

Инструменты

Какие есть инструменты для генерации тестовых данных

- faker

Инструменты

Какие есть инструменты для генерации тестовых данных

- faker
- falso

Инструменты

Что предлагают эти инструменты нам?

- Возможность генерить данные, используя предметную область

Инструменты

Что предлагают эти инструменты нам?

- Возможность генерить данные, используя предметную область
- Человеко-читаемые и понятные тестовые данные, а не набор рандомных букв и цифр

Инструменты

Что предлагают эти инструменты нам?

- Возможность генерить данные, используя предметную область
- Человеко-читаемые и понятные тестовые данные, а не набор рандомных букв и цифр
- Возможность получать всегда разные, но при этом валидные данные (хорошо для списков / таблиц и т.д.)

Тестовые данные

person

```
import { faker } from '@faker-js/faker';

export function getPerson() {
  return {
    firstName: faker.name.firstName(),
    lastName: faker.name.lastName(),
    addresses: [
      {
        city: faker.address.city(),
        street: faker.address.streetAddress(),
        appartment: faker.datatype.number(),
      }
    ]
  }
}
```

Тестовые данные

person

```
import { randCity, randFirstName, randLastName, randNumber, randStreetAddress }
from '@ngneat/falso';

export function getPerson() {
  return {
    firstName: randFirstName(),
    lastName: randLastName(),
    addresses: [
      {
        city: randCity(),
        street: randStreetAddress(),
        appartment: randNumber({ min: 1, max: 99 }),
      }
    ]
  }
}
```

Тестовые данные

response

```
export function mockInternationalContract(): InternationalContractV3 {
  return {
    id: randSequence({ size: 43 }),
    number: randAccount({ accountLength: 5 }),
    date: formatDateTime(randRecentDate({ days: 150 })),
    uniqueNumber: randMask({ mask: '########/####/####/#/#' }),
    clientIds: [randAccount()],
    name: randSentence(),
    status: rand(Object.values(InternationalContractV3.StatusEnum)),
    registrationDate: formatDateTime(randRecentDate({ days: 150 })),
    closureDate: formatDateTime(randRecentDate({ days: 150 })),
    amount: randAmount({ min: 100, max: 10000 }),
    currencyCode: rand(mockCurrencies).code,
    balance: randAmount({ min: 100, max: 10000 }),
    completionDate: formatDateTime(randFutureDate({ years: 2 })),
    advance: randBoolean(),
    currencyClause: { type: 'CENTRAL_BANK_RATE' },
    periodicFixedPayments: randBoolean(),
    paidAmount: randAmount({ min: 100, max: 10000 }),
    fulfilledAmount: randAmount({ min: 100, max: 10000 }),
    prolongation: randBoolean(),
  };
}
```

Инструменты

Рублёвые платежи

Импорт | Создать платеж

Исходящие | Входящие | Шаблоны | Картотека

№	Дата	Контрагент	Назначение	Сумма	Статус
85063	04.10.2022	Dooley and Sons 40302.450.8.79538886425	Оплата по договору 2354-3	8 224 203.97 RUB	Исполнен
1571	03.10.2022	Weber Inc 40302.714.3.15645208888	Оплата по договору 2354-1	621 345.91 RUB	Создан
54018	02.10.2022	Weber Inc 40302.563.1.85823432403	Оплата за материалы по договору №25-Р от 29.03.2020 и накладной №156 от...	3 103 236.22 RUB	Требуется подтверждение по СМС
90633	30.09.2022	Welch, Lockman and Hand 40302.717.1.47470448786	Оплата по договору 2353 НДС не облагается	2 202 192.23 RUB	Черновик
63624	30.09.2022	Kuhlman, Schowalter and West 26468.465.6.87605469343	Предварительная оплата за транспортные услуги по счёту № 12 от...	6 477 357.93 RUB	Создан
48703	29.09.2022	Jacobi - Kutch 40302.874.3.94423783487	Оплата по договору 2354-3	3 150 731.67 RUB	Отменен
84122	28.09.2022	Steuber, Luettgen and Corkery 72861.638.0.84316833175	Оказание услуг по договору 432 на основании акта № 25 от 25 июня 2020...	2 904 416.21 RUB	В работе

Инструменты

Кредиты

Создать заявку

Кредиты Заявления

Договор	Ставка %	Сумма кредита	Следующий платёж	Задолженность
Возобновляемая кредитная линия MOS/RK/345/88 от 22.11.2021	12	15 891 705.72 RUB	5 584 453.34 RUB —	5 772 476.93 RUB
Овердрафт MOS/RK/298/04 от 08.06.2022	13	18 347 521.67 RUB	11 319 657.47 RUB —	11 264 000.12 RUB
Овердрафт MOS/RK/415/51 от 21.06.2022	13.5	18 866 544.16 RUB	5 523 875.21 RUB —	11 389 444.02 RUB
Транш MOS/RK/456/76 от 28.08.2022	13.5	15 311 192.20 RUB	13 819 489.10 RUB —	12 085 236.22 RUB
Возобновляемая кредитная линия MOS/RK/133/48 от 05.05.2022	10	18 209 434.72 RUB	7 835 323.29 RUB —	12 238 290.58 RUB
Кредитный договор MOS/RK/678/72 от 29.11.2021	10	15 643 385.56 RUB	7 352 859.54 RUB —	7 217 222.16 RUB

Faker

Тестовые данные

person

```
import { faker } from '@faker-js/faker';

export function getPerson() {
  return {
    firstName: faker.name.firstName(),
    lastName: faker.name.lastName(),
    addresses: [
      {
        city: faker.address.city(),
        street: faker.address.streetAddress(),
        appartment: faker.datatype.number(),
      }
    ]
  }
}
```

Faker

плюсы и минусы

- Faker — самый популярный инструмент

Faker

плюсы и минусы

- Faker — самый популярный инструмент
- Достаточно простой API

Faker

плюсы и минусы

- Faker — самый популярный инструмент
- Достаточно простой API
- Развивается сообществом (форк от автора — неадекватного мейнтенера)

Faker

плюсы и минусы

- Faker — самый популярный инструмент
- Достаточно простой API
- Развивается сообществом (форк от автора — неадекватного мейнтенера)
- Хорошо с переводами на разные локали

Faker

плюсы и минусы

- Faker — самый популярный инструмент
- Достаточно простой API
- Развивается сообществом (форк от автора — неадекватного мейнтенера)
- Хорошо с переводами на разные локали
- Имеет не tree-shakeable структуру (все от объекта faker)

Faker

плюсы и минусы

- Faker — самый популярный инструмент
- Достаточно простой API
- Развивается сообществом (форк от автора — неадекватного мейнтенера)
- Хорошо с переводами на разные локали
- Имеет не tree-shakeable структуру (все от объекта faker)
- Большой размер библиотеки тянется весь из-за отсутствия tree-shaking

Falso

Тестовые данные

person

```
import { randCity, randFirstName, randLastName, randNumber, randStreetAddress }
from '@ngneat/falso';

export function getPerson() {
  return {
    firstName: randFirstName(),
    lastName: randLastName(),
    addresses: [
      {
        city: randCity(),
        street: randStreetAddress(),
        appartment: randNumber({ min: 1, max: 99 }),
      }
    ]
  }
}
```

Falso

плюсы и минусы

- Создатель Falso — известный мейнтейнер Netanel Basal

Falso

плюсы и минусы

- Создатель Falso — известный мейнтейнер Netanel Basal
- Достаточно простой и tree-shakeable API

Falso

плюсы и минусы

- Создатель Falso — известный мейнтейнер Netanel Basal
- Достаточно простой и tree-shakeable API
- Малый размер бандла из-за использования только того, что нужно

Falso

плюсы и минусы

- Создатель Falso — известный мейнтейнер Netanel Basal
- Достаточно простой и tree-shakeable API
- Малый размер бандла из-за использования только того, что нужно
- Плохо с переводами на разные локали (вы можете пополнить локали - PRы приветствуются)

Falso

плюсы и минусы

- Создатель Falso — известный мейнтейнер Netanel Basal
- Достаточно простой и tree-shakeable API
- Малый размер бандла из-за использования только того, что нужно
- Плохо с переводами на разные локали (вы можете пополнить локали - PRы приветствуются)
- Менее известный инструмент

Итоги

Выводы

- Нужно взаимодействовать с бэкендерами до реализации

Выводы

- Нужно взаимодействовать с бэкендерами до реализации
- Можно двигаться в разработке параллельно с бэкендерами, а не последовательно

Выводы

- Нужно взаимодействовать с бэкендерами до реализации
- Можно двигаться в разработке параллельно с бэкендерами, а не последовательно
- Можно показывать работу на моковых данных продактам

Выводы

- Нужно взаимодействовать с бэкендерами до реализации
- Можно двигаться в разработке параллельно с бэкендерами, а не последовательно
- Можно показывать работу на моковых данных продактам
- Понятные тестовые данные для тестов / деморежима

Выводы

- Нужно взаимодействовать с бэкендерами до реализации
- Можно двигаться в разработке параллельно с бэкендерами, а не последовательно
- Можно показывать работу на моковых данных продактам
- Понятные тестовые данные для тестов / деморежима
- Простота перехода от моков к реальным данным

Вопросы?



